text. This object result file is the file **43** according to FIG. **3** which is displayed at the browser. In the illustrated example, the multiple DIS capsule data retrieval command file **91**(*a*) . . . **91**(*n*) initiates as a first step multiple queries to different databases which are specified by the parameters of the request. In this example, multiple queries are initiates as SQL type search requests as multiple steps **91**(*a*) . . . **91**(*n*) executed by the DIS capsule server **133** with the Database Gateway **134** to select data from DB26000 databases located inside the intranet **140** and on the Internet by Internetwork routing to database gateway **134'** and its DB26000 databases by step **91**(*a*). The data is stored in a DIS declared buffer. Similarly, in parallel or successively, additional steps **91**(*b*), **91**(*c*, **91**(*d*), and **91**(*n*) retrieve data and store in their object buffer data retrieved from Sybase, Oracle, Redbrick, and IBM's Data Warehouse databases. Thus object **91**(*a*) will query DB26000 and bring data back to DIS. Object **91**(*b*) will query Oracle and bring data back to DIS. Object **91**(*c*) will query Sybase and bring data back to DIS. Object **91**(*d*) (shown as a dot in FIG. **9**) will query Redbrick and bring data back to DIS, and so on. The nth object **91**(*n*) will query IBM's data warehouse and bring data back to DIS. In a subsequent linked processing step **92** data from the database queries in the first step is joined by joining object command file **92** and stored in a buffer related to this object. Object **92** will join the data from the n locations searched in step **91**. Thereafter, in a subsequent processing step performed by calculation object command file **93** on the joined data in the joined database result buffer of step **92**, desired calculations performed in accordance with the parameters indicated by the request are done on the joined data. Thereafter, in accordance with the request parameters text is formatted to space delimited text by the format object command file **94**. The results are stored in a buffer associated with format object command file **94**. Thereafter, a make text command file **95** causes the formatted text to be created as a text file for the WWW server **131** to be stored in a file which is accessible to and can be retrieved and displayed by the control program agent **73**, or directly displayed by the control program agent **73** in the form illustrated in FIG. **4** at the Web browser **130**. It will be noted we have illustrated this process as object capsules in a DIS internetworking environment. These object capsules are a specialized form of a command file, which can encompass additional commands called by an object.

Preferred Embodiment of graphics DIS capsule

FIG. **10** illustrates by way of example a DIS capsule that creates a graphical report file. For simplicity, data in this FIGURE is also shown in a DIS environment **90**. Retrieval object command file **101** illustrates a step of retrieval of data from one or more databases as specified in the parameters of the request, performing these retrieval steps as did retrieval object command files **91**(*a*) . . . **91**(*n*). Thereafter, this data is plotted with the make plot object command file **102**, with the results being stored in a buffer. The final step of creating a result-to-be-presented file, in this instance in the form of a bitmap ready for display to a Web browser **130** is created by the make bitmap (BMP) object command file **103**. The example of a preferred bitmap object command which would be employed with todays Internet environment is a GIF image. Others can be used as well. Again the results are provided to the Web browser **130**, by the action of the program command agent **73** on the Web Server **131**, the results being illustrated by the pie-chart of FIG. **6** in accordance with the parameters of the request for generating the graphical report illustrated by FIG. **6**.

Alternative Preferred Embodiments

FIG. **11** illustrates an alternative configuration of the network system as it may be employed for permitting access to information available through homepages and in data warehouses where access to the homepage or database may or may not be restricted by a firewall. In FIG. **11**, the web browser(s) **130** accesses an associated Web Server **131**,**131'**, **131"** either by a coupling or addressing with a uniform resource locator (URL) the Web Server **131** which may be selected with a Hypelink. This can be a direct coupling or an indirect coupling, as via a node locatable in a common access medium, such as provided by Internet resources accessible via a web browser, e.g. supporting Web Explorer, or Mosaic, NetScape, node **131** located somewhere on the Internet which utilizes our control program agent **73**. Now node **131** which functions as a Web server is coupled via a token-ring network, SNA network, or other suitable network **132** (one of the any which may be used on the Internet as a transmission medium) with the facilities provided within what we will call our intranet, those facilities which are "proprietary" to the owner and which may be protected by firewalls at the intranet boundary **140**. Now note that our control program **73** is resident within the Web Server **131** and functions as described in FIG. **8** to couple to a DIS server **133** located within the intranet **140**, which is preferably located behind a firewall as indicated in FIG. **11**. This DIS Server **133** is in turn coupled to our Database gateway **134**. This database gateway is configured as illustrated also in FIG. **1** for gathering information from databases coupled to it and located on servers for DB2, Oracle, Sybase, and Redbrick, as well as one for information warehouse functions. In our preferred embodiments these database units are IBM mainframe systems, as available commercially today, but they could be AS400s, RISC/6000, RISC/6000 SP or other systems supporting the databases.

The DIS Server is a server which supports DIS or similar decision support functions and the functions provided by our DIS capsules illustrated by FIG. **9** and **10**.

Now our Web browsers **130** can not only access information within the intranet, but can reach outside the intranet to gather information located elsewhere via the Internet. We will describe two examples of our preferred couplings to elements on the Internet. One example couples the database gateway **134** to another (a second) database gateway **134'** via the Internet and it Internetwork routing (INR) protocol available from IBM as part of its current DIS product which can make use of UALs. The second database gateway **134'** is coupled to its own (second) DIS server **133'**. At this point the Web browser **130** can access data not only intranet, but also via the Internet to gather data from a database supported by DIS server **133'** located outside the intranet. The Database server **134'** would be able to gather information from any database coupled to it, as illustrated, assuming access is public or accessible after processing of a hidden variable access authorization.

However, the web browser(s) **130** can also access via Web Server **131** (with our control program **73** illustrated in detail in FIG. **8**) another Web server **131'** which implements our control program **73**. This Web server, for example, Web server **131'** can also be coupled via its own (second) network **132'** (which supports functions equivalent to network **132** and as illustrated in FIGS. **1** and **11**) to an associated DIS Server **133'** as illustrated to perform tasks like those we are describing from a request sent via the second network from its Web server **131'**.

However, as another alternative example, Web server **131'** with an appropriate API can access a directly coupled